NPS67-83-02CR

NAVAL POSTGRADUATE SCHOOL

Monterey, California

CONTRACTOR REPORT

A FORTRAN PROGRAM FOR SOLVING TWO-DIMENSIONAL EULER EQUATIONS WITH GODUNOV METHODS -USER'S MANUAL

S. Eidelman

421 Dela Vina Apt. C
Monterey, CA 93940

February 1983

Contractor Report

Prepared for:
Naval Postgraduate School
, California 93940

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Commodore R. H. Shumaker
Superintendent

D. A. Schrady
Provost

The work reported herein was carried out for the Turbopropulsion Laboratory (TPL) by the author under Contract Number N62271-83-M-0977. The report serves to provide preliminary documentation of a new computer code developed while the author was an NRC Research Associate in the NPS Foundation Research Program. The author's work contributes to a broad effort at TPL to develop improved methods for transonic turboaerodynamics, and was funded by the Airbreathing Propulsion Research Program of Naval Air Systems Command, Code AIR 310.

This report was prepared by:

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NPS67-83-02CR	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): A Fortran Program for Solving Two-Dimensional Euler Equations with Godunov Methods-User's Manual	5. TYPE OF REPORT & PERIOD COVERED: Contractor Report January 1983-February 1983
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): S. Eidelman	8. CONTRACT OR GRANT NUMBER(s): N62271-83-M-0977
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Shmuel Eidelman, 421 Dela Vina Apt. C, Monterey, CA 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Postgraduate School, Monterey, CA 93943	12. REPORT DATE: February 1983
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computational Fluid Dynamics

Euler Equations Solvers

Flow in Transonic Compressor

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A description is given of a program which solves the two-dimensional Euler equations using Godunov Methods. The program was recoded in Fortran from Pascal and accomodates either cascade or channel flow boundary conditions. The equations, boundary conditions and numerical method are outlined and the structure of the program is given. Results using the Fortran and Pascal versions are briefly compared. The documentation describes a program undergoing development and is not a finished guide for general use.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

For many analysis problems in turbomachinery, an assumption of inviscid flow provides sufficiently accurate results for design or development tasks. At the same time the assumption or irrotational flow will limit application to low speed shockless cases. The Euler equations describe inviscid rotational flow and therefore allow the modeling of transonic or supersonic flows with shock waves and rotational flow fields behind strong waves. Solutions of the Euler equations are therefore important for many applications.

A computer program was developed which solves numerically two-dimensional Euler equations using the Godunov method (Ref. 1) or Higher Order Extension of the Godunov method (Ref. 2). The program was intended primarily for turbomachinery applications, but with modification of the boundary conditions, application to any problem which is modeled by the Euler equations is possible.

The purpose of this report is to give instructions to potential users of the present computer code and to outline the basic assumptions made in the mathematical model and numerical solution.

II. GOVERNING EQUATIONS

The unsteady two-dimensional Euler equations can be written in conservation law form as

$$\frac{\partial \bar{a}}{\partial t} + \frac{\partial \bar{b}}{\partial x} + \frac{\partial \bar{c}}{\partial y} = 0 \qquad (1)$$

where

$$\bar{a} = \begin{vmatrix} \rho \\ \rho U \\ \rho V \\ e \end{vmatrix}, \quad \bar{b} = \begin{vmatrix} \rho U \\ p + \rho U^2 \\ \rho UV \\ (e + p)U \end{vmatrix}, \quad \bar{c} = \begin{vmatrix} \rho V \\ \rho UV \\ p + \rho V^2 \\ (e + p)V \end{vmatrix} \qquad (2)$$

and

$e = \rho(\varepsilon + \frac{U^2 + V^2}{2})$ = energy of the unit of volume

$\varepsilon = \frac{p}{(\gamma - 1)\rho}$ - internal energy

ρ = density

U and V are the velocity components corresponding to the x and y coordinates

p = pressure

γ = specific heat ratio

Equation (1) is non-dimensionalized using a reference length p_o, a reference pressure x_o and a reference density ρ_o. These reference quantities are in principle arbitrary and were selected as follows

x_o = relative length of the geometrical cord

$p_o = 101350.0 \frac{N}{m^2}$

$\rho_o = 1.2 \text{ kg/m}^3$

The reference quantities for time, t_o and velocity, a_o are not independent and are determined from

$$\frac{a_o\, t_o}{x_o} = 1 \; ; \quad \frac{p_o}{\rho_o\, a_o^2} = 1 \; ; \tag{3}$$

then

$$a_o = \sqrt{\frac{p_o}{\rho_o}}$$

$$t_o = x_o / a_o \; ;$$

non-dimensionalize the variables in Eq. (1) as follows;

$$\begin{aligned} U &= a_o\, U_n \; ; \; p = p_o\, p_n \; ; \\ \rho &= \rho_o\, \rho_n \; ; \; V = a_o\, V_n \; ; \\ x &= x_o\, x_n \; ; \; y = x_o\, y_n \; ; \\ t &= t_o\, t_n \end{aligned} \tag{4}$$

When these quantities are inserted in Eq. (1), the flow equations, in terms of non-dimensional variables, are identical in form to those in terms of dimensional variables. The subscript "n" may then be dropped without confusion and only non-dimensional quantities need be referred to in the remaining sections of this report. We will solve Eq. (1) in physical space, so no further transformation of the equation is required.

III. BOUNDARY CONDITIONS

The computational domain illustrated in Figure 1 is bounded by open surfaces (flow may cross these surfaces), solid surfaces and inflow-outflow boundaries. Formulation and implementation of boundary conditions at each surface are discussed in this section.

3.1. SOLID WALL BOUNDARY CONDITION

The solid wall boundary condition appropriate for inviscid flow computations is that of zero mass flux through the surface. This condition is difficult to implement uniquely for the Euler equations.

In the program developed, the user has a choice of two types of boundary conditions on the solid walls as follows;

a) Solution of the Riemann problem between the point of the domain of integration and its mirror image in the direction normal to the wall with the same parameters but, with an inverse sign of velocity component normal to the wall. (See Fig. 2).

b) The wall pressure is derived from the momentum equation

$$\frac{\partial p}{\partial n} = \frac{\rho U_p^2}{R_s} \qquad (5)$$

where

n - indicates the direction normal to the surface

U_p = velocity parallel to the surface

R_s = radius of curvature of the surface

If we add the flow tangency condition, we have a uniquely defined boundary condition at the wall. The solid wall boundary conditions are imposed at the cascade surface, which corresponds to the segments 3-5 and 4-6 in the computational domain shown in Figure 1.

3.2. PERIODIC BOUNDARIES

The periodic flow boundary conditions are imposed on the segments 1-3, 5-7, 2-4, and 6-8 in Fig. 1. By virtue of the periodicity of the flow with respect to y, the flow parameters over segments 1-3 and 5-7 are the same as those over 2-4 and 6-8. Hence, the segments 1-3, 5-7, 2-4, and 6-8 are essentially internal, there is no need to apply any boundary conditions to them. In the numerical solution additional cells are attached to but outside these regions as shown in Figure 1. The flow parameters in the extra cells are set equal to the flow parameters in the corresponding cells inside the computational domain which are displaced in the Y direction by the cascade blade spacing, S.

3.3. INLET CONDITIONS

The U component of velocity, the flow angle θ, the gas stagnation enthalpy, H, and the entropy, S, are kept constant at the inlet. This leads to a unique definition of the flow parameters at the inlet boundary as follows

$$U_{in} = \text{const}; \quad V_{in}/U_{in} = \tan\theta = \text{const};$$

$$\frac{\gamma\, p_{in}}{(\gamma - 1)\, \rho_{in}} + \frac{U_{in}^2 + V_{in}^2}{2} = H = \text{const}; \tag{6}$$

$$\frac{p_{in}}{\rho_{in}^{\gamma}} = S = \text{const};$$

The explicit formula for V_{in}, ρ_{in}, and P_{in} are therefore

$$V_{in} = U_{in} \tan\theta$$

$$\rho_{in} = \left[\frac{(\gamma-1)}{\gamma S}\left(H - \frac{U_{in}^2 + V_{in}^2}{2}\right)\right]^{\frac{1}{\gamma-1}} \tag{6.1}$$

$$p_{in} = S\, \rho_{in}^{\gamma}$$

3.4. OUTLET CONDITIONS

At the outlet boundary we define only pressure p_{out} and for all other flow parameters apply continuation conditions. This means that U_{out}, V_{out}, and ρ_{out} are set equal to the values of U, V and ρ one point ahead of the outlet boundary.

IV. THE NUMERICAL METHOD

In the program there is a choice of using one of the following numerical methods:

a) First order accurate Godunov Method (Ref. 1)

b) Higher order extension of the Godunov method (Ref.2)

The higher order method is an extension of the Godunov method and was used as it is described in Ref. 2. The piecewise-linear algorithm was used rather than piecewise-parabolic.

Here we will describe the Godunov method as it is implemented by the program.

Let us assume that the grid is covering the computational domain as is shown in Fig. 2(a). We will also consider a fixed grid with the indexing of lines as shown in Fig. 2(a). The flow parameters related to the center of the cell will have a fractional index: $j + 1/2$, $k + 1/2$. The cell boundaries will have one fractional and one integer index: $j, k + 1/2$ or $m + 1/2, n$. The parameters at the time t will have subscript index (i.e., $\rho_{j + 1/2, k + 1/2}$) in distinction from the parameters at the time $t + \Delta t$ which will have superscript index: $\rho^{j + 1/2, k + 1/2}$

In this notation Eq. (1) is approximated to the first order of accuracy by the following system of equations in finite differences:

$$\bar{a}^{\,j+\frac{1}{2},k+\frac{1}{2}} = \bar{a}_{j+\frac{1}{2},k+\frac{1}{2}} - \frac{\Delta t}{\sigma_{j+\frac{1}{2},k+\frac{1}{2}}} \cdot$$

$$(B_{j+\frac{1}{2},k} \cdot \Delta Y_{j+\frac{1}{2},k} - C_{j+\frac{1}{2},k} \cdot \Delta X_{j+\frac{1}{2},k} +$$

$$B_{j+\frac{1}{2},k+\frac{1}{2}} \cdot \Delta Y_{j+1,k+\frac{1}{2}} - C_{j+1,k+\frac{1}{2}} \cdot \Delta X_{j+1,k+\frac{1}{2}} +$$

$$C_{j+\frac{1}{2},k+1} \cdot \Delta X_{j+\frac{1}{2},k+1} - B_{j+\frac{1}{2},k+1} \cdot \Delta Y_{j+\frac{1}{2},k+1} +$$

$$C_{j,k+\frac{1}{2}} \cdot \Delta X_{j,k+\frac{1}{2}} - B_{j,k+\frac{1}{2}} \cdot \Delta Y_{j,k+\frac{1}{2}}) \qquad (7)$$

where

$$\sigma_{j+\frac{1}{2},k+\frac{1}{2}} = \tfrac{1}{2} \quad (X_{j+1,k+1} - X_{j,k})(Y_{j,k+1} - Y_{j+1,k}) -$$

$$(X_{j,k+1} - X_{j+1,k})(Y_{j+1,k+1} - Y_{j,k})$$

$$\Delta Y_{j,k+\frac{1}{2}} = Y_{j,k+1} - Y_{j,k}$$

$$\Delta X_{j+\frac{1}{2},k} = X_{j+1,k} - X_{j,k}$$

$$\bar{a} = \begin{vmatrix} \rho \\ \rho U \\ \rho V \\ e \end{vmatrix} \quad , \quad \bar{B} = \begin{vmatrix} RU \\ P+RU^2 \\ RUV \\ (E+P)U \end{vmatrix} \quad , \quad \bar{C} = \begin{vmatrix} RU \\ RUV \\ P+RV^2 \\ (E+P)V \end{vmatrix}$$

Here, and in the following text with capital letters R, U, V, P, etc., we will note the parameters calculated at the edges of the zones. To obtain the value of these parameters we calculate the Riemann problem with left and right states in the centers of adjacent cells (Ref. 3). For example, to calculate the flow parameters on the edge $j,k+\frac{1}{2}$ we assume that the left state is located in $j-\frac{1}{2},k+\frac{1}{2}$ and the right state in $j+\frac{1}{2},k+\frac{1}{2}$ (see Fig. 2). The velocity vectors $(U,V)_{j-\frac{1}{2},k-\frac{1}{2}}$ and $(U,V)_{j+\frac{1}{2},k+\frac{1}{2}}$ should be expressed in terms of normal and tangent components of velocity to the edge (j,k+1) - (j,k). Then the Riemann problem is solved for following parameters:

$$\begin{aligned} &\text{Left state: } (\rho,\ P,\ U_n,\ V_t)_{j-\frac{1}{2},k+\frac{1}{2}} \\ &\text{Right state: } (\rho,\ P,\ U_n,\ V_t)_{j+\frac{1}{2},k+\frac{1}{2}} \end{aligned} \tag{8}$$

where

U_n - the velocity component normal to the edge (j,k+1)-(j,k)

V_t - the velocity component tangent to the edge (j,k+1)-(j,k)

Solution of this Riemann problem will give:

$$(R,\ P,\ U_n,\ V_t)_{j,k+\frac{1}{2}}$$

Then we transform the component of the velocity back to the cartesian coordinate system and will have the final value of the parameters on the cell edge:

$(R, P, U, B)_{j,k+\frac{1}{2}}$

That is a brief description of the Godunov method. A more detailed description of the method can be found in Reference 4.

The program includes higher order extension of the Godunov method proposed by Collela (Ref. 6) which, in order to improve accuracy, adds the following steps before calculating the Riemann problem:

1) Compute linear profiles of the dependent variables in each zone by interpolating slopes at the centers of the zones, subject to certain monotonicity constraints. This gives rise to a global distribution of the dependent variables which is piecewise linear, linear in each zone, with jump discontinuities at the edges of the zones.

2) Correct left and right states by tracing approximate characteristics and solving difference approximations to the characteristic equations. Only then is the Riemann problem solved for the corrected right and left states and the values obtained by this solution are assigned to the corresponding edge of the grid cell.

V. COMPUTER PROGRAM

The program described here was developed on the NPS IBM 370-3033 computer.

Since the computer code was first developed in Pascal, which implies structured programming, the program structure is simple. The Data Flow Diagram in Fig. 3 illustrates the general flow of data in the program and the subroutine functions. The diagram is self-explanatory.

5.1. GENERAL DESCRIPTION

The computer program consists of 18 routines. Here we will describe, in short, the general function of each routine. To understand fully the place of each routine and the program structure, the reader should follow the Data Flow Diagram in Fig. 3 while reading the written description.

The subroutine MAIN calls all other subroutines in the program. It controls the execution of the program by checking two constants ITRNUM and ITRCH. ITRNUM is the number of iterations performed. ITRCH controls the number of iterations from the last call to the write routines (WRVELC, WRPRES, WRDENS or WRINTG). Three important constants are defined in MAIN:

MAX = number of grid points in x direction

MAY = number of grid points in y direction

LEV = intergration level. If LEV=10 the integration will be on two grid levels (not recommended). If LEV=9 it is only one level of integration.

Subroutine RDDATA reads the grid definition data which is created on File 3 by the successive execution of the programs

GRID and TRNEW. The program GRID creates a two dimensional grid of the H-type for the given cascade definition. The program TRNEW redefines the grid to the form needed for application of the method and calculates the cosine and sine of angles defining the direction normal to the cell edge. Program TRNEW also defines the array REG which simulates the computational domain by assigning to each physical point in the domain an integer number. The description of the programs GRID and TRNEW will not be given in the present manual. Here we shall assume that File 3, for a given cascade, has been created. This file contains the following data:

REG - array of integer numbers which simulates the structure of the computational domain

A1 - array containing Δy values on the cell edges and the area of the cell located in the point $(j+\frac{1}{2},k+\frac{1}{2})$

B1 - array containing Δx values on the cell edges

TCOS - cosine of the angle between the normal direction to the cell edge and direction of the x-axis

TSIN - sine of the angle between the normal direction to the cell edge and direction of the x-axis

VF - x coordinates of the points on the grid covering the computational domain

VY - y coordinates of the points on the grid

If the program continues integration from a certain step for which the values of the flow parameters are known, these values should be stored on File 4 (program creates new data on File 4 after a certain number of iterations). In this case

subroutine RDDATA reads three additional arrays from File 4:

RO - density

P - pressure

UB - velocity component in x direction

VB - velocity component in y direction

Subroutine RDDATA defines the following constants:

ITNO - number of iteration in the solution of the Riemann problem

RZ - characteristic density

PZ - characteristic pressure

XZ - characteristic length

G - gamma

Subroutine DEFINE defines the following parameters:

ITRNUM - number of iteration

TIMTOT - total time

RADL - array, radii of curvature on the lower side of the profile

RADUP - array, radii of curvature on the upper side of the profile

If the program is executing from the beginning, the intial values of the array RO, P, UB, VB are defined in the subroutine DEFINE. The initial enthalpy - ENTALP, entropy - ENTROP and tanθ - TAU are calculated in this subroutine.

Subroutine TIME calculates the time step for the next integration based on CFL criterion -TJ. This subroutine writes the values of ITRNUM, TJ, TIMTOT, MOMENT, ans MASS where

MOMENT - integral of momentum in x direction

MASS - integral of mass

Subroutine INOUT defines the inlet and outlet boundary conditions for the flow parameters. In this routine we calculate the values of the flow parameter in the points (2,k) and (MAX-1,k), where k has the values corresponding only to the centers of the cells.

Subroutine BOND defines boundary conditions for the computational domain, except inlet and outlet boundary conditions which are defined in the routine INOUT. Two kinds of boundary conditions are calculated in this routine, as a function of the cell type indicator REG(j,k):

1) Boundary condition at the upper and lower surfaces of the cascade profile

2) Symmetry boundary conditions for the regions 1-3, 5-7, 2-4, and 6-8. To calculate the values of the flow parameters at the cell edges this subroutine calls routine ITER to solve the Riemann problem at the edges.

Subroutine RIMSOL defines and solves the Riemann problem along k lines between the centers of the cells. This routine has an option to calculate the more exact values for the left and right states calling routine CHARY. A condition statement in this routine controls call of the routine CHARY that allows the use of the same code for the Godunov method and its higher order accurate extension.

Subroutine RIMSLX defines and solves the Riemann problem at the edges of the cells in the y line directions. In the case of higher order accurate integration, this routine should

call the subroutine CHAR. That call is controlled by a condition statement.

Subroutine INTGR performs integration of Eq. (1) in accordance with the numerical formula (7). This subroutine calculates the values of MOMENT, MASS and ENERGY.

Subroutine WRVELC calculates and writes the values of the local Mach number for the center of each cell on File 6.

Subroutine WRPRES writes the values of the normalized pressure P(j,k) in the centers of the cells on File 6. As an option this subroutine calculates and writes the pressure coefficient C_p instead of P(j,k).

Subroutine WRDENS is included as optional and it writes the values of the array RO(j,k) (normalized density) on File 6.

Subroutine UUB writes the values of the local Mach number corresponding to the profile surface on File 6.

Subroutine WRINTG writes complete content of the arrays: RO, P, UB AND VB, on File 4. That allows in the next execution of the program, beginning integration from the last iteration of the previous execution of the program.

Subroutine VISCO is optional and is not used in regular computation. This routine adds the numerical viscosity of Lapidus type (Ref. 5) to the conservation equations.

Subroutine RIEMAN calculates the Riemann problem for the defined left and right states.

Subroutine ITER (called only from subroutine RIEMAN) calculates pressure on the interface for the solution of the Riemann problem.

Subroutine CHAR corrects values of the left and right states solving the equations of gas dynamics in the characteristics form. In CHAR this correction is done in the x direction.

Subroutine CHARY is the same as CHAR, only for y direction.

Subroutine MONO, is called only form CHAR or CHARY. This subroutine computes linear profiles of the dependent variables in each zone by interpolating slopes at the centers of the zones, subject to certain monotonicity constraints.

VI. INPUT

All external input to the program is written on File 3. Data on File 3 is written in Format (26I3) for array REG(j,k) and in Format (5F15.12) for all other arrays. If the user wants to continue iteration from the last iteration of the previous execution of the program, then the user should input the values of the arrays of the flow parameter. This data is recorded on File 4 with Format (5F15.12).

All external input arrays are read from the subroutine RDDATA. The inlet parameters of the flow and the outlet pressure are defined in the subroutine DEFINE from line 1560 to 1710.

VII. OUTPUT

All output of the program are recorded on File 6 and File 4. The following subroutines write output on to File 6:

TIME, WRVELC, WRPRES, WRDENS, UUB.

Only subroutine WRINTG writes on File 4.

VIII. RESULTS

The Fortran version of the computer code was tested for a number of problems for which the solution was obtained with the Pascal version. The conclusions from this testing are:

1) Fortran program is ≃50% faster than Pascal program, when executed with the option Optimize (2) on the Fortran H compiler with single precision.

2) Execution with Fortran G compiler is 30% faster than Pascal.

3) Execution with Fortran H compiler with Optimize (1) option is 30% faster than Pascal.

4) Execution with Fortran H compiler with double precision with Optimize (2) option is only 25% faster than Pascal.

5) The differences in results are caused by different levels of precision involved in the calculations.

IX. REFERENCES

1. Godunov, S. K. et al, "Numerical Solution of Multidimensional problems in Gas Dynamics," Nauka, 1976 (in Russian).

2. Colella, P. and Woodward, P. R., "The Piecewise-Parabolic Method (PPM) for Gas-Dynamical Simulations," in press Journal of Computational Physics, 1983.

3. Chorin A. J., Journal of Computational Physics, 23, p. 517, (1976).

4. Eidelman, S., Colella, P. and Shreeve, R. P., "Application of the Godunov Method and Higher Order Extension of the Godunov Method to Cascade Flow Modeling," AIAA Paper-83-1941-CP, AIAA 6th Computational Fluid Dynamics Conference, 1983.

5. Lapidus, A., Journal of Computational Physics, 2, p. 154, (1976)

6. Colella, P., private communication.

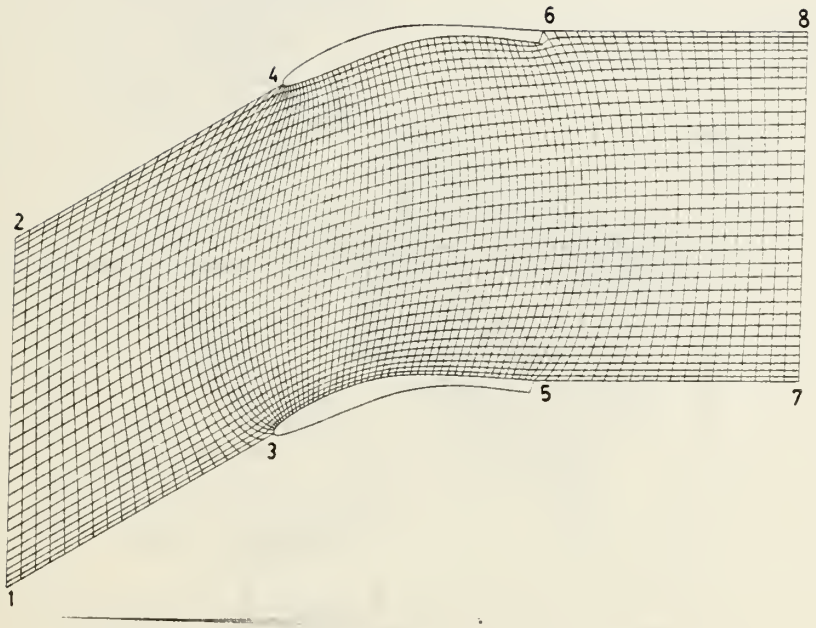
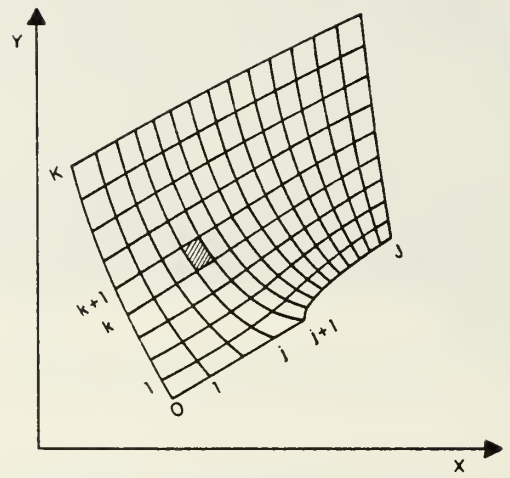


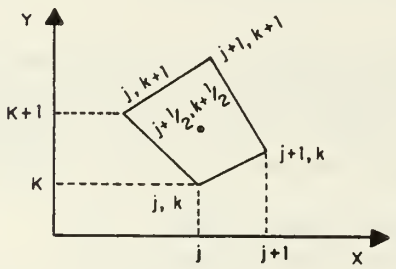
Fig. 1. Computational grid and boundary of the computational domain.

We look for the solution of the system of equations represented by equation (1) in the computational domain shown in Fig. 1 for $t \to +\infty$ with the following conditions at the domain boundaries:

a) Inflow along segment 1-2
b) Outflow along segment 7-8
c) Solid wall along segments 3-5 and 4-6
d) Cascade: Periodicity between segments 1-3 and 2-4 and between segments 5-7 and 6-8
   Channel: Solid wall along segments 1-7 and 2-8



a) Grid Location in the Coordinate System



b) Single cell

Fig. 2. Computational Space Notation.

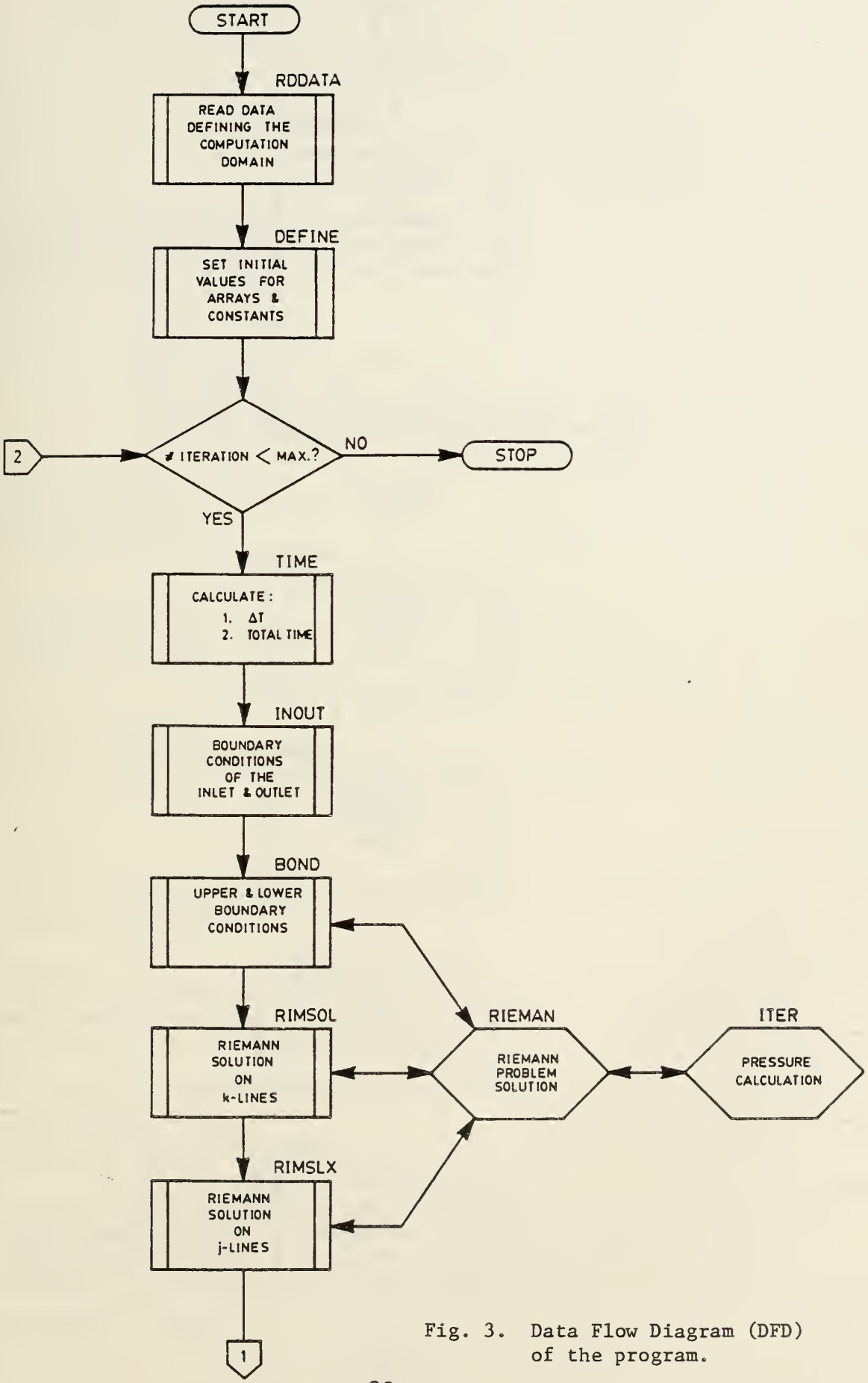


Fig. 3. Data Flow Diagram (DFD) of the program.

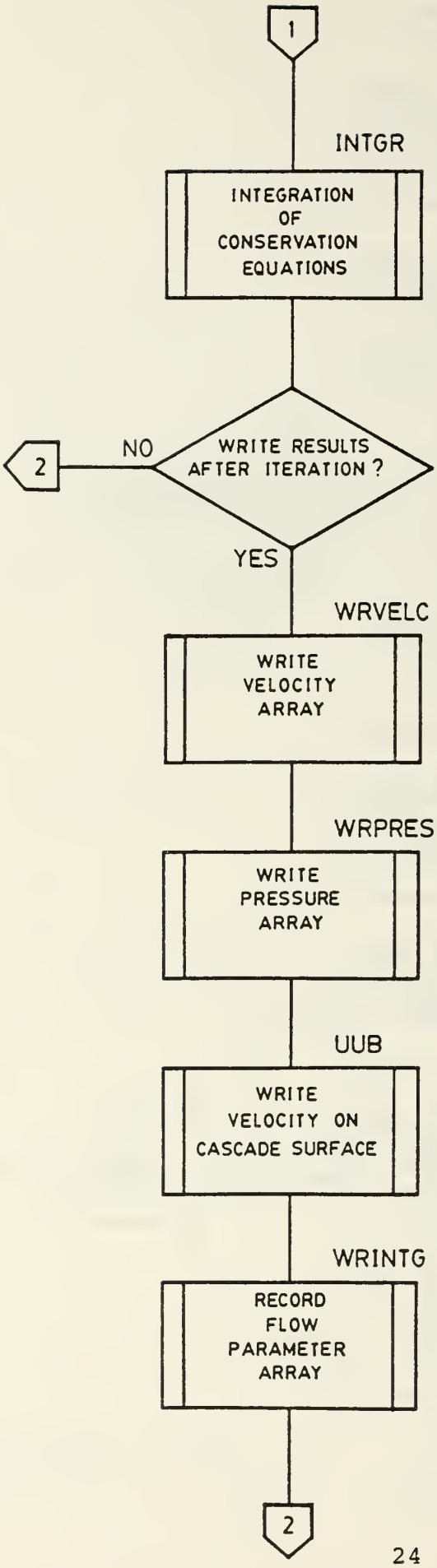


Fig. 3. (continued)

DISTRIBUTION LIST

		No. Copies
1.	Commander Naval Air Systems Command Washington, D.C. 20361 Attention: Code AIR 310	2
	Code AIR 310E	1
	Code AIR 310F	1
	Code AIR 320D	1
	Code AIR 530	1
	Code AIR 536	1
	Code AIR 00D4	1
	Code AIR 03D	1
2.	Office of Naval Research 800 N. Quincy St. Arlington, VA 22217 Attention: Dr. A. D. Wood	2
3.	Commanding Officer Naval Air Propulsion Center Trenton, NJ 08628 Attention: T. Elsasser	2
4.	Commanding Officer Naval Air Development Center Warminster, PA 19112 Attention: AVTD	1
5.	Library Code 1424 Naval Postgraduate School Monterey, CA 93940	4
6.	Library Army Aviation Material Laboratories Department of the Army Fort Eustis, VA 23604	1
7.	Dr. Arthur J. Wennerstrom AFWAL/POTX Wright-Patterson AFB Dayton, OH 45433	1
8.	National Aeronautics & Space Administration Lewis Research Center (Library) 21000 Brookpark Road Cleveland, OH 44135	1

9. Library 1
General Electric Company
Aircraft Engine Technology Division
DTO Mail Drop H43
Cincinnati, OH 45215

10. Library 1
Pratt & Whitney Aircraft Group
P.O. Box 2691
West Palm Beach, FL 33402

11. Library 1
Pratt & Whitney Aircraft Group
East Hartford, CT 06108

12. Chief, Fan and Compressor Branch 1
Mail Stop 5-9
NASA Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44135

13. Library 1
Air Research Mfg. Corporation
Division of Garrett Corporation
412 South 36th Street
Pheonix, AZ 85034

14. Library 1
Curtiss Wright Corporation
Woodridge, NJ 07075

15. Library 1
AVCO/Lycoming
550 S. Main Street
Stratford, CT 06497

16. Library 1
Williams International
P.O. Box 200
Walled Lake, MI 48088

17. Library 1
Teledyne CAE, Turbine Engines
1330 Laskey Road
Toledo, OH 43612

18. Library 1
Detroit Diesel Allison Division G.M.C.
P.O. Box 894
Indianapolis, IN 46202

19. Library 1
The Garrett Corporation
AIResearch Manufacturing Company
402 S. 36 Street
Phoenix, AZ 85034

20. Turbopropulsion Laboratory 15
Code 67Sf
Naval Postgraduate School
Monterey, CA 93943

21. NPS Research Administration 1
Code 0121
Naval Postgraduate School
Monterey, CA 93943

22. Professor Shmuel Eidelman 2
Code 67Em
Naval Postgraduate School
Monterey, CA 93943